Resource Scheduling for a Network of Communications Antennas

S. Chien, R. Lam, and Quoc Vu
Jet Propulsion Laboratory
4800 Oak Grove Drive
Pasadena, CA 91109-S099"
{steve.chien,raymond.lam,quoc.vu}@jpl.nasa.gov

Abstract- This paper describes the Demand Access Network Scheduler (DANS) system for automatically scheduling and rescheduling resources for a network of communications antennas. DANS accepts a baseline schedule and supports rescheduling of antenna and subsystem resources to satisfy tracking goals in the event of: changing track requests, equipment outages, and inclement weather.

TABLE OF CONTENTS

1. INTRODUCTION

The Deep Space Network (DSN) [2] was established in 1958 and since then it has evolved into the largest and most sensitive scientific telecommunications and radio navigation network in the world. The purpose of the DSN is to support unpiloted interplanetary spacecraft missions and support radio and radar astronomy observations in the exploration of the solar system and the universe. There are three deep space communications complexes, located in Canberra, Australia, Madrid, Spain, and Goldstone, California. Each DSN complex operates four deep space stations -- one 70-meter antenna, two 34-meter antennas, and one 26-meter antenna. The functions of the DSN are to receive telemetry signals from spacecraft, transmit commands that control the spacecraft operating modes, generate the radio navigation data used to locate and guide the spacecraft to its destination, and acquire flight radio science, radio and radar astronomy, very long baseline interferometry, and geodynamics measurements.

From its inception the DSN has been driven by the need to create increasingly more sensitive telecommunications devices and better techniques for navigation. The operation of the DSN communications complexes require a high level of manual interaction with the devices in the communications link with the spacecraft. In more recent times NASA has added some new drivers to the development of the DSN: (i) reduce the cost of operating the DSN, (2) improve the operability, reliability, and maintainability of the DSN, and (3) prepare for a new era of space exploration with the New Millennium program: support small, intelligent space.craft requiring very few mission operations personnel.

The purpose of this paper is to describe The DANS system for rescheduling **and** resource allocation for antenna and subsystem resources in the DSN. DANS works from an initial schedule and uses prioritized pre-emption and localized search to find antenna and other equipment resources required to support changes to schedule requirements which may be caused by a wide range of circumstances including: changing track

This work was performed by the Jet Propulsion Laboratory, California Institute of Technology, under contract with the National Aeronautics and Space Administration.

requirements from the flight projects, equipment outages, and inclement weather.

This paper is organized in the following manner. We begin by characterizing the current mode of operations of the DSN. Next we describe the architecture automation of multiple levels of 'I'X operations. In particular, we describe the role of and relationship between: the Demand Access Network Scheduler (DANS) system for automated resource allocation, DPLAN [3,4], for automatically constructing tracking plans, and the NMC, a plan execution and monitoring system. In addition we provide examples of the inputs and outputs to each of the components to illustrate what occurs at each step in the process of capturing spacecraft data[1].

2. HOW THE DSN OPERATES

Voyager-1 is cruising at 17.5 kilometers/second toward the outer edge of the solar system. Though its onboard systems are mostly asleep during this phase of its mission, Voyager's health metrics are continually sent to Earth via a telemetry signal radiated by its 40-watt transmitter. It will take eight hours at the speed of light for the signal to reach its destination, Earth, a billion miles away. Upon arrival, the telemetry signal is received by an extremely sensitive ground communications system, NASA's Deep Space Network (DSN), where it is recorded, processed, and sent to the Mission Operations and Voyager project engineers, who assess the health of the spacecraft based on the contents of the signal.

The type of activity just described occurs daily for dozens of different NASA spacecraft and projects that use the DSN to capture spacecraft data. Though the process of sending signals from a spacecraft to Earth is conceptually simple, in reality there are many earthside challenges that must be addressed before a spacecraft's signal is acquired and transformed into useful information.

[1] For a further description of DSN antenna operations as at I applications are for planning and plan execution see (Chien et al. 1996)

Network Preparation at the Network Operations Control Center

The first stage of the DSN tracking process is called Network Preparation and it occurs at a central control center for the DSN located at JPL called the Network Operations Control Center (NOCC). The project initiates Network Preparation by sending a request for the DSN to track a spacecraft involving specific tracking services. The DSN responds to the request by attempting to schedule the resources (i.e., an antenna and other shared equipment) needed for the track.

Along with this request, the project prepares a Sequence of Events (SOE) describing the time-ordered activities that should occur during the track. The SOE includes actions that the DSN should take, (e.g., begin tracking the project's spacecraft at 1200" hours), and it also includes events that will occur on the spacecraft being tracked (e.g., the spacecraft will change frequency or mode at a designated time). These events are important because they affect how the DSN provides the services. The project SOE is sent to the DSN, which then generates its own version, called a Ground Network SOE. The Ground Network SOE is a more elaborate version of the project SOE in that it expands the activities from high level descriptions (e.g., begin tracking the spacecraft) into a finer level of detail for use by the operations personnel at the deep space station. The Ground Network SOE is sent to the Deep Space Station (DSS), where the antennas used to perform the actual track are located. Along with the Ground Network SOE a wide range of required support data are transmitted - such as the predicted location of the spacecraft, etc.

Data Capture At The Signal Processing Center

The data capture process is performed by operations personnel at the deep space station - they determine the correct steps to perform to configure the equipment for the track, perform the actual establishment of the

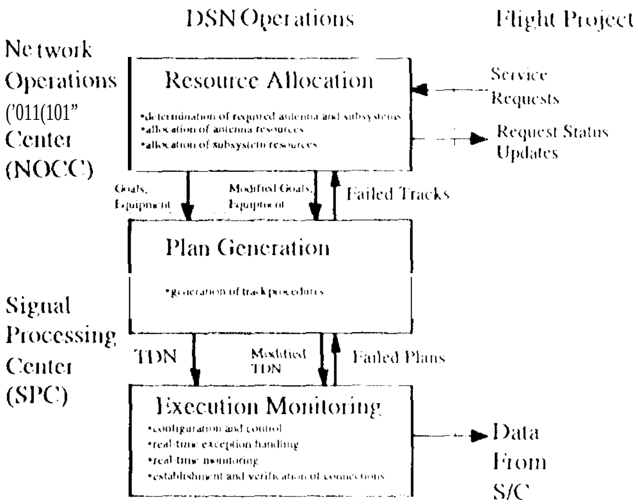


Figure 1: An Automation Oriented View of Deep Space Network Operations

communications link, which we hereafter refer to as a 'link', and then perform the track by issuing control commands to the various subsystems comprising the link. Throughout the track the operators continually monitor the status of the link and handle exceptions (e. g., the receiver breaks lock with the spacecraft) as they occur. All of these actions are currently performed by human operators, who manually issue tens or hundreds commands via a computer keyboard to the link subsystems. The monitoring activities require the operator to track the state of each of the subsystems in the link (usually three to five subsystems), where each subsystem has many different state variables that change over time.

Automation Systems

In the last section we described the current labor-intensive process for transforming a flight project service request into an executable set of DSN operations. Efforts are currently underway to automate aspects of these tasks. In particular, DANS is being developed to automate aspects of the resource scheduling process, DPLAN is being fielded to automate generation of the tracking plans, and the NMC system is being fielded to automatically execute the operations procedures.

3. AUTOMATED SCHEDULING OF DSN RESOURCES - OVERVIEW

Each week, a complex matching process between spacecraft project communication service requests and NASA Deep Space Network (DSN) resources occurs. In this process, project requests and priorities are matched up with available resources in order to meet communications needs for earth-orbiting and deep space spacecraft. This scheduling process involves considerations of thousands of possible tracks, tens of projects, tens of antenna resources and considerations of hundreds of subsystem configurations. Once this initial schedule is produced (8 or more weeks before implementations), it undergoes continual modification due to changing project needs, equipment availability, and weather considerations. Responding to changing context and minimizing disruption while rescheduling is a key issue.

This paper describes the Demand-based Automated Network Scheduling System (DANS), an automated scheduling system being developed at the Jet Propulsion Laboratory (JPL) to schedule DSN resources. DANS uses priority-driven, best-first, constraint-based search and iterative optimization techniques to perform priority-based rescheduling in response to changing network demand. In this techniques, DANS first considers the antenna allocation process, as antennas are the central focus of resource contention. After establishing a range of antenna options, DANS then considers allocation of the 5-13 subsystems per track (out of the tens of shared subsystems at each antenna complex) used by each track. DANS uses constraint-driven, branch and bound, best first search to efficiently consider the large set of possible subsystems schedules.

The high level resource allocation problem for the smaller DSN antennas (26M and smaller) is currently handled by the OMP scheduler. In the future, an evolution of OMP called the Demand Access Network Scheduler (DANS) scheduling system is designed to address rescheduling for the entire DSN (e.g., all antennas and antenna subsystems). OMP accepts generalized service requests 11"011"1 spacecraft projects of the form "we need three 4-110111' tracks per week" and resolves conflicts using a priority request scheme to attempt to maximize satisfaction of high priority projects. OMP deals with sched ules for NASA's 26-meter subnet involving thousands of possible tracks and a final schedule involving hundreds of [sacks.

4. DOMAIN CHARACTERISTICS

In addition to the basic antenna resource allocation probl em, the DSN scheduling problem is further complicated by three factors: (1) context-dependent priority; (2) subsystem allocation; and (3) the possibility of reducing the length of the tracks. DSN track priorities are context dependent in that they are often contingent on the amount of tracking the project has received so far in the week. I 'or example, a project might have priority 3 to get 5 tracks, priority 4 to get 7 tracks and priorit y 6 to get 9 tracks (where lower priorities represent mm e important tracks). This reflects that 5 tracks are necessary to maintain spacecraft health and get critical science data to ground stations; 7 tracks will allow a nominal amount of science data to be downlinked; and 9 tracks will allow for downlinking of all science data (e.g., beyond this level additional tracks have little utility). An important point is that specific tracks are not labeled with these priorities (e. g., t he project is allowed to submit 5 tracks at priority 3, 2 at priority 4 and so OH). Rather when considering adding, deleting, 01 moving tracks the scheduler 11111s1 consider the overall priority of the project in the current alloc ation context.

In addition to allocating antennas, DSN scheduling involves allocating antenna subsyste ms which are shared hy each Signal Processing Center (such as telemetry J)rocessor" transmitters and exciters). Allocating these complicates the scheduling problem because it adds to the number O f resources being scheduled and certain subsystems may only be required for parts of the track.

Finally, the DSN scheduling problem is complicated by the fact that the track duration can be relaxed. For example, a project may request a 3 hour track but specify a minimum track time of 2, hours. When evaluating potential I esource conflicts the scheduler must consider the option of shortening tracks to remove resource conflicts. Currently OMP and J)ANS use a linear weighting scheme in conjunction with a modified SIMPI .EX algorithm to trim tracks in accordance with prioritizations.

1)ANS accepts two types of inputs: 1 , an 8-week prior- to operation schedule from the I<c.source Allocation and Planning (RAP) team, and 2, acti Vity requests from each individual flight project. The 8- week schedule is the baseline for creating a conflict-free schedule. Many of the scheduled activities at that tir ne are tentative at best, and therefore subject to revision due to changing p roject status. Also, the schedule is for the antenna resources only; I).SN subsy stem scheduli ng is not considered at all in the 8-week schedule.

The activity requests are used by the flight projects to add and delete activities on an existing sched ule due to changing project requirements and/or resource availability. The 1)ANS objective is to satisfy as many activity requests as possible while maintaining a cmlfiict-flee stat us with min imum disruption to the existing schedule.

J)ANS uses many data structur c classes to represent the DSN domain, including timel ines, projects, activities, and many others to support the domain knowledge and reasoning pro cess. Based 011 the request as well as resource constrai nts, DANS reasons with the domain knowledge, employs appropriate strateg ies, considers different alternatives, and eventuali y generates a conflict- free schedule for all the. antenna and subsystem resources. I)ANS is intended for use by the operation personnel to maintain and update the J)SN schedule throughout each schedul e.

The DSN domain contains many resources. In the existing configuration (as of July

1996), it consists of 11 signal processing centers (SPC), 45 antennas, and 161 subsystems. They are located at different sites around the world. The majority of the antennas can be classified as 26, 34, and '70 meter antennas. The 26 meter antennas on average handles 600 activities per week. The 34 and 70 meter antennas performs over 200 activities per week. Additionally, this workload is expected to increased dramatically in the next several years.

The DSN domain has many unique attributes. During the mission design and pre-planning stages (many years prior to a spacecraft launch), projects identify items such as uplink and downlink frequencies, major events which will occur during the life of the spacecraft, and the expected life of the spacecraft. This information is used to create a long-term resource schedule (8 weeks or more). During DSN routine operation, regular communications between the ground stations and the spacecraft are required to command the spacecraft, to monitor spacecraft status, and to collect scientific data. These activities are submitted to and modified on the near-team schedule (8 weeks or less) on a regular basis.

Another issue is the placement of activities onto the schedule. The possible times for a spacecraft track are limited by spacecraft orbit views, which are the periods in which the spacecraft is visible from a ground station. Also, the range from the antenna to spacecraft dictates the quantity and types of antenna(s) required for each activity. Sometimes, an array of multiple antennas instead of a single one is required to communicate with the spacecraft. In addition, the uplink and downlink activities can occur on different antennas, and can be several hours apart imposing additional dependencies between activities.

There are two types of activities in the DSN domain: spacecraft activities and ground activities. Spacecraft activities are submitted by projects and used to interact with spacecraft. They are required to satisfy the domain constraints above. Ground activities represent hardware maintenance. Antenna time which is not occupied by spacecraft activities is used for ground activities such as non- regular maintenance requirements and testing, with maintenance having higher priority.

Each DSN spacecraft activity is divided into 3 steps: pre-calibration (precal), tracking, and post-calibration (postcal). The time periods for each step are specified in ranges of values. The time periods are unique for each activity type, and depend on the antenna type and subsystem usage. DANS models this dependency by either shrinking or shifting activities to maximize resource utilization as dictated by activity type, antenna type, and subsystem usage.

DANS is required to schedule two different kinds of DSN resources: antennas and subsystems. Antennas and subsystems are Unit resources and as such can not be shared by more than one activity. Subsystem resources are hardware such as transmitters which are required to work with an antenna during communication. Normally, there are many pieces of hardware that support each antenna and DANS is required to generate a schedule which allocates all necessary subsystems.

5. PROBLEM REPRESENTATION

The following data structures are used in DANS to represent the DSN domain information, and for supporting the inference process: capacity timeline, project, and pass classes. These data structures are described below.

Capacity Timeline Class

The DSN consists of many SPCs situated around the globe. Each SPC may contain one or more antennas. The antenna sizes range from 9 meters for communicating with low earth orbit spacecraft to 70 meters for communicating with deep space spacecraft. In addition to the antennas on each SPC site, there are many subsystems associated with the SPC and the antennas. The current physical arrangement of the DSN makes it natural to represent these resources in a hierarchical manner as shown in Figure 1. At the top of the hierarchical tree is a SPC resource. Each SPC contains one or more antennas, which are the children of the SPC. There are also many SPC subsystem resources residing also as children of the SPC. For each antennas in the hierarchy tree, there are many DSS subsystems associated with it. To further illustrate this hierarchy, Figure 2. shows the hierarchy for the SPC-10 DSN complex.

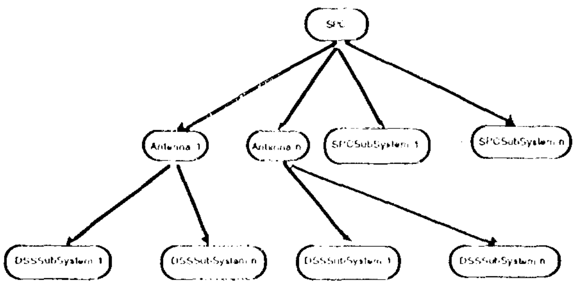


Figure 1: DSN Resources Hierarchy

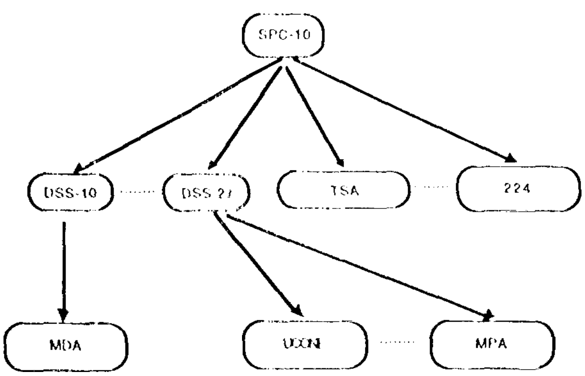


Figure 2: SPC-10 Resource Hierarchy Example

All the DSN resources are represented by the capacity timeline class (CptyTimeline). The CptyTimeline models a resource's usage at any instant of time for the duration of a schedule. It is composed of one or more instances of the capacity unit class (CptyUnit). A CptyUnit is used to represent a constant resource usage within a time period. Shown in Figure 3 is the timeline representation of the DSS-26 antenna resource.

Since the CptyTimeline represents the state of a resource for the schedule's duration, it is the most often used data structure in the system. The CptyTimeline is constantly being modified and updated during the inference process to reflect the state of the resource at that instant of time. In order to increase the system performance, a time slice caching scheme is used to expedite the query process. The scheme equally divides the timeline into a number of buckets. During a query, the system will have to find the bucket that contains the moment first, then it will search sequentially within that bucket to match the query. For example, when the system looks for a CptyUnit which contains the 1:30 moment as shown in Figure 3, it first identifies Bucket #2 as the container which includes the 1:30 moment. Then it traverses down the timeline starting from the beginning of Bucket #2 until it locates the CptyUnit that contains the 1:30 moment.

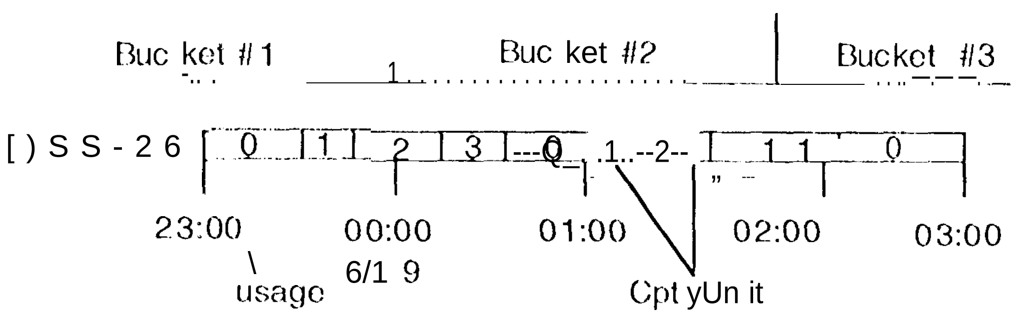


Figure 3: Capacity Timeline Resource Representation

6. INFERENCE ENGINE

The current version of DANS employs a priority-based inference strategy to satisfy the project requests. Future versions will incorporate other strategies to make the reasoning process more robust.

The major bottleneck of the DSN domain is the antenna resources. This is due to the fact that there are limited numbers of antennas available, and they can only commit to one activity in any given time. On the other hand, there are many identical subsystem resources that support unexpected events. Subsequently, the scheduling of the subsystems is still important but less critical. Using a top-down approach, the inference engine can focus on conflict resolution of critical resources first. Then, it can reason at the next level of abstraction to resolve less-critical resource conflicts. The Priority-Based strategy exploits this representation to reason at both the antenna and the subsystem levels.

The scheduling process is separated into three major steps. First, the system generates an exhaustive list of solutions for each activity request at the antenna level. Second, this solutions list is then applied to the subsystem level to pick the best solution. Finally, the activity is placed onto the schedule. If the activity addition causes another activity deletion, the deleted activity(ies) will be rescheduled immediately. For ground subsystem maintenance activities which do not require antenna resources, the scheduling process will skip the first step(antenna scheduling) and start from the second step(subsystem scheduling). The antenna inferencing and subsystem inferencing flowcharts are shown in Figures 4 and 5.

At the DSN antenna inferencing level, the system first selects an activity request (ACT) and then validates the window in which the ACT can exist on the schedule. This window is then mapped into the spacecraft orbit views in order to identify the intervals at which a specific antenna can view the spacecraft. The result is a list of one or more valid intervals within the window. For each possible interval, the system tries to schedule the ACT into it. When conflict arises between the ACT and the current schedule, the system first tries to shift the ACT within the interval. If this action does not resolve the conflict and the conflicting activities have lower priority, the system identifies the conflicted activity (ies) for deletion. If a solution exists, then the committed time slot, conflicting activities, along with the cost index is stored in a list for later comparison. After the system identifies all possible solutions at the antenna level, it sorts the solutions based on the cost index so that the best solution will be placed at the beginning of the list.

At the DSN subsystem inferencing level, DANS first identifies all the subsystems which are required to support the ACT. Then DANS selects the first available antenna solution, and tries to schedule the ACT to each of the subsystems at the specified time slot.

If conflict exists, it will try to resolve it as described above. When a solution exists, the system calculates the completed solution cost for both the antenna(s) and subsystems, and compares them to the previous completed solution costs and next antenna solution. If the current solution cost is less than or equal to the other solution costs, the system will commit to this solution, and will schedule the ACT to this specified time slot. Otherwise, this solution" will be saved for future reference.

After the system evaluates all the antenna solutions at the subsystem level, it will pick the best solution with the lowest cost to schedule the activity request. If this action requires deletion of other lower prioritized activities, the deleted activities will be submitted back to the schedule as a request immediately.

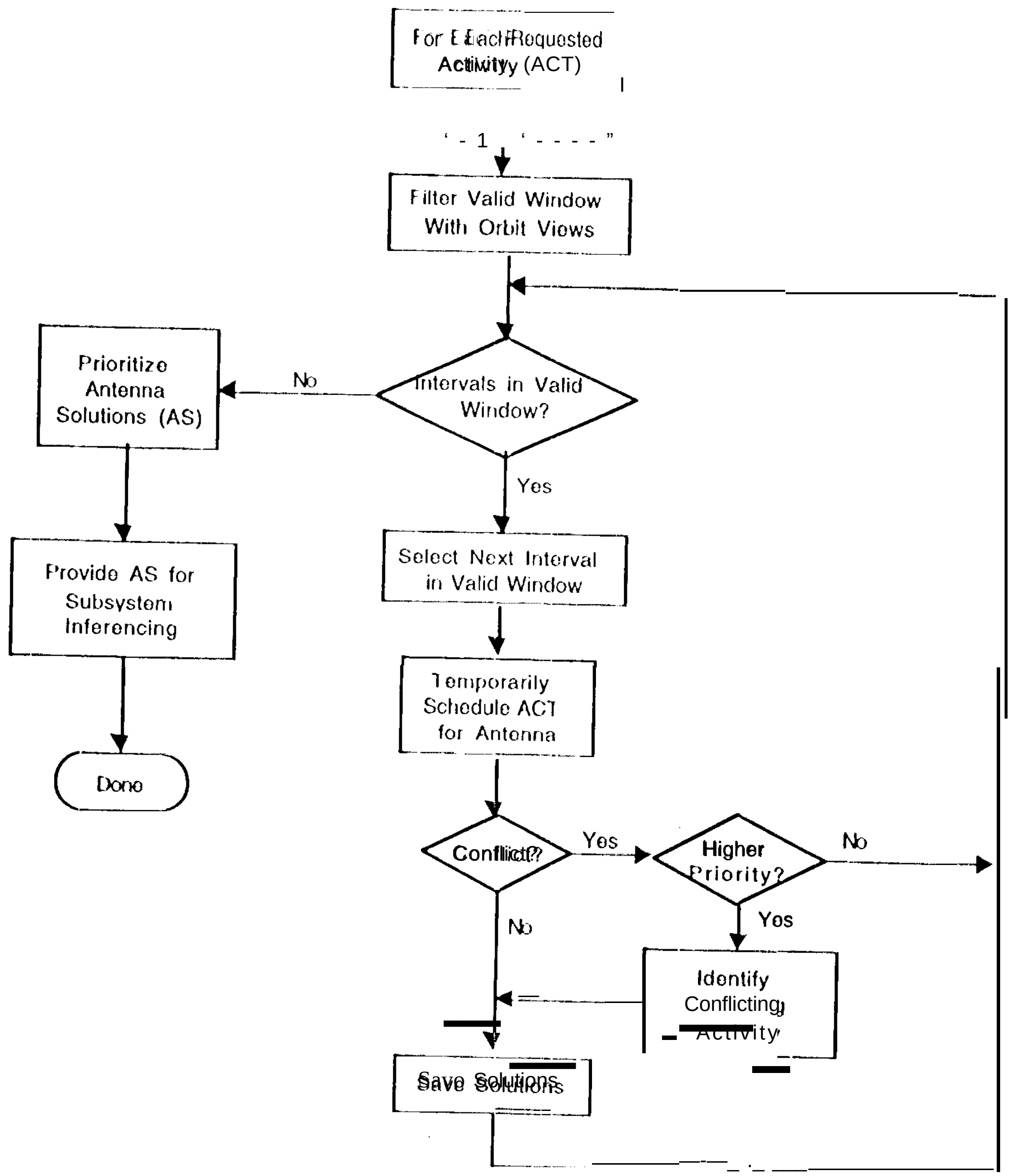


Figure 4: Priority-Based Antenna Inferencing Flowchart

DANS uses an equation to calculate each solution's cost. The equation is as follows:

$$\text{Solution Cost} = \frac{\text{NAD} * \text{activity priority}}{\text{NAD} - (\text{NAD} - 1) * 0.1}$$

where NAD = number of deletions required to schedule the current activity.

The cost is basal on the activity priority. When there is no deletion required for scheduling the ACT, the cost is zero. When one deletion is required, the cost is equal to the activity priority. When there is more than one deletion required in order to schedule the ACT, the cost increases with the. number of deletions.

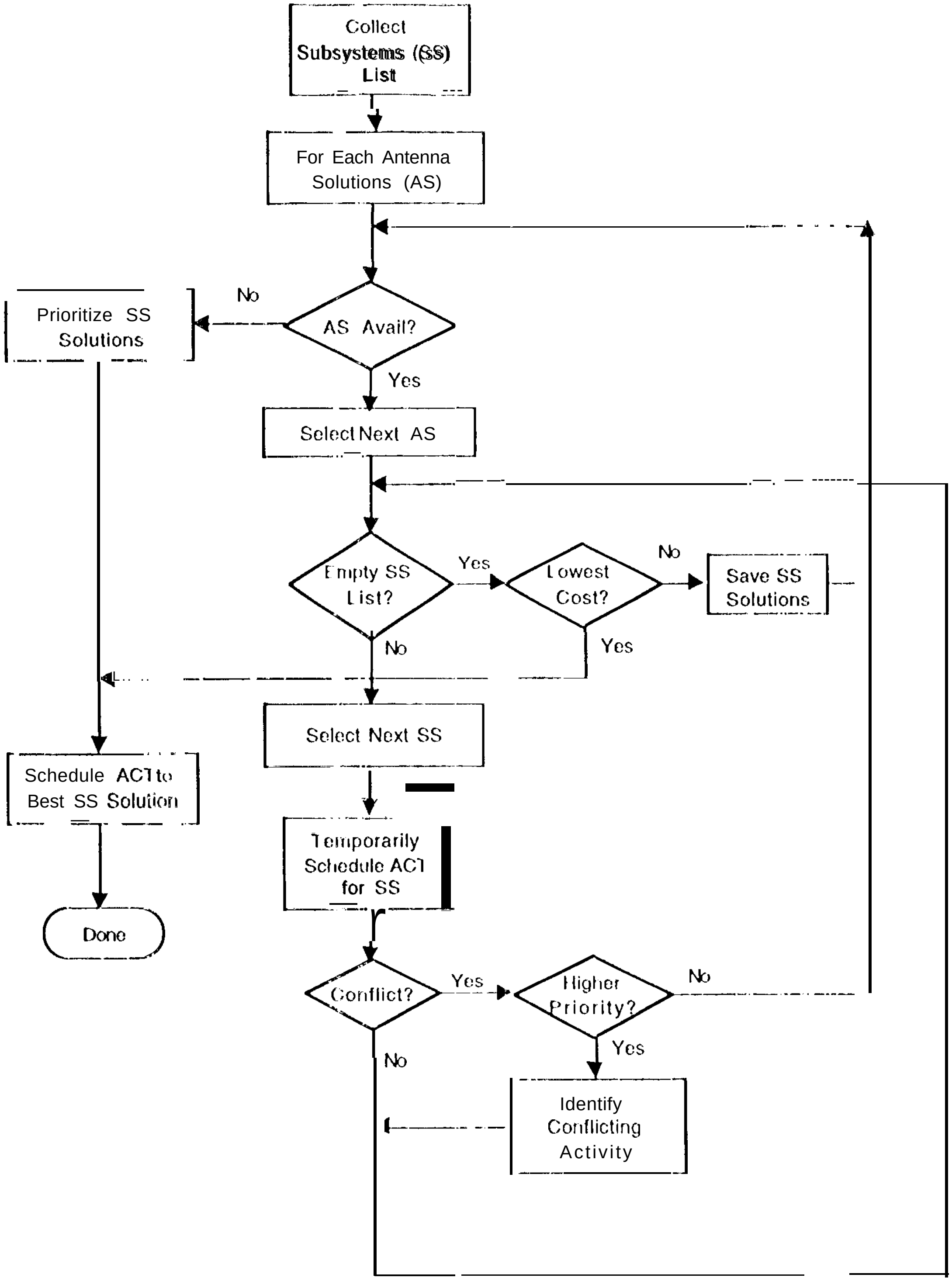


Figure 5: Priority-Based Subsystem Inferencing Flowchart

7. PRIORITY-BASED RESCHEDULING: AN EXAMPLE

This section provides an example of how DANS places an activity request into an existing schedule.

Initially, the DSS-14 antenna and its subsystems have committed their resources to two activities between 6:00am and 10:15am. Activity P0 has a valid window from 7:45am to 12:45pm, and occupies the 7:45am to 8:45am time slot. Activity P1 has a valid window from 9:15am to 12:45pm, and occupies the 9:15am to 10:15am time slot. Both P0 and P1 are DSN ground activities with priorities equal to 4. Activity P6 is a Galileo activity which requests a two hour duration between 5:45am and 10:15am on the DSS-14 resources. P6 has a priority value of 3, which is higher than the priority of both P0 and P1. Subsequently, P6 can bump these two activities from the timelines when conflict arises. This information is shown in Table 1

The DANS objective is to commit DSS-14 and subsystem resources to P6 activity and to maintain the conflict-free schedule with minimum disruption to the existing schedule. The DANS scheduling process involves hypotheses generation, conflict identification, and conflict resolution. See Figure 6 for the scheduling sequences for this example. Appendix C contains the actual DANS output for this example.

For the Galileo P6 request, DANS first identifies all orbit views which are subsets of the P6 valid window. This enables DANS to filter out the invalid gaps and limits the search space. For this example, there is only one orbit view existing from 6:00am to 10:00am.

Then the system turns its attention to the critical antenna resource to generate hypotheses. It traverses within the valid orbit view duration on the DSS-14 antenna timeline to identify time slots which can satisfy the 2 hour duration constraint. There are two valid time slots: Time Slot 1 from 6:00am to 8:45am; and Time Slot 2 from 7:45am to 10:00am. DANS schedules P6 to both Time Slot 1 and Time Slot 2 to create two hypotheses. When P6 is place at Time Slot 1 for hypothesis 1 (HY1), it causes conflict with P0. Since P6 has higher priority than both P0 and P1, placement of P6 within this duration will delete activity P0 and the antenna solution cost becomes 4. When P6 is placed at Time Slot 2 for hypothesis 2 (HY2), it causes deletion of both P0 and P1, and the antenna solution cost is 4.21053. The system then sort all the hypotheses based on the antenna solution cost in ascending order. The result guides the inference process at the subsystem level without the necessity of performing an exhaustive search.

The system then continues the conflict identification at the subsystem level for both hypotheses. Activity P6 requires seven subsystem resources to accomplish the task. They are the LMC, SLE, TGC-A, MDA, NAR, RCV, and S-TWM. DANS identifies resource conflicts with P0 for the LMC, MDA, RCV subsystems for both hypotheses. The combined solution cost for the HY1 becomes 8.28571. This combined cost is then compared to the HY2 antenna cost, which is 4.21053. If the combined cost would have been less than the antenna cost value, the system would stop here and select the current hypothesis as the best solution. Since this is not the case, the system continues on the next hypothesis and calculates the combined cost for HY2 as 8.53485.

Activity	Project	Priority	Orbit View	Request Duration	Valid Window	Assignment
P0	DSN	4	N/A	60	7:45-12:45	9:15-10:15
P1	DSN	4	N/A	60	9:15-12:45	9:15-10:15
P6	GLLO	3	6:00-10:00	120	[illegible]15	

Table : Example Activities Description

Based on the result, DANS selects the first hypothesis as the solution since it has the lowest combined cost. It schedules 1'6 to the 6:00am to 8: 00am duration and deletes 1 ' 0 from the resource timelines. The system applies the same process to reschedule 1'0 activity. It identifies 3 time slots to generate hypotheses as shown in Figure 6. The system identifies the first time slot to schedule P0 between 8:00am and 9:15am with zero cost. It stops here having completed its task successfully to place the Galileo activity on the timeline without deleting any existing activities from the schedule.

8. RESCHEDULING CONTEXT

At the high level of resource allocation, schedule execution dots not involve execution monitoring. However, rescheduling is often necessary due to: equipment outages, last minute track requests, last minute changes to scheduled tracks, and atmospheric conditions impact on tracking capabilities. Rescheduling

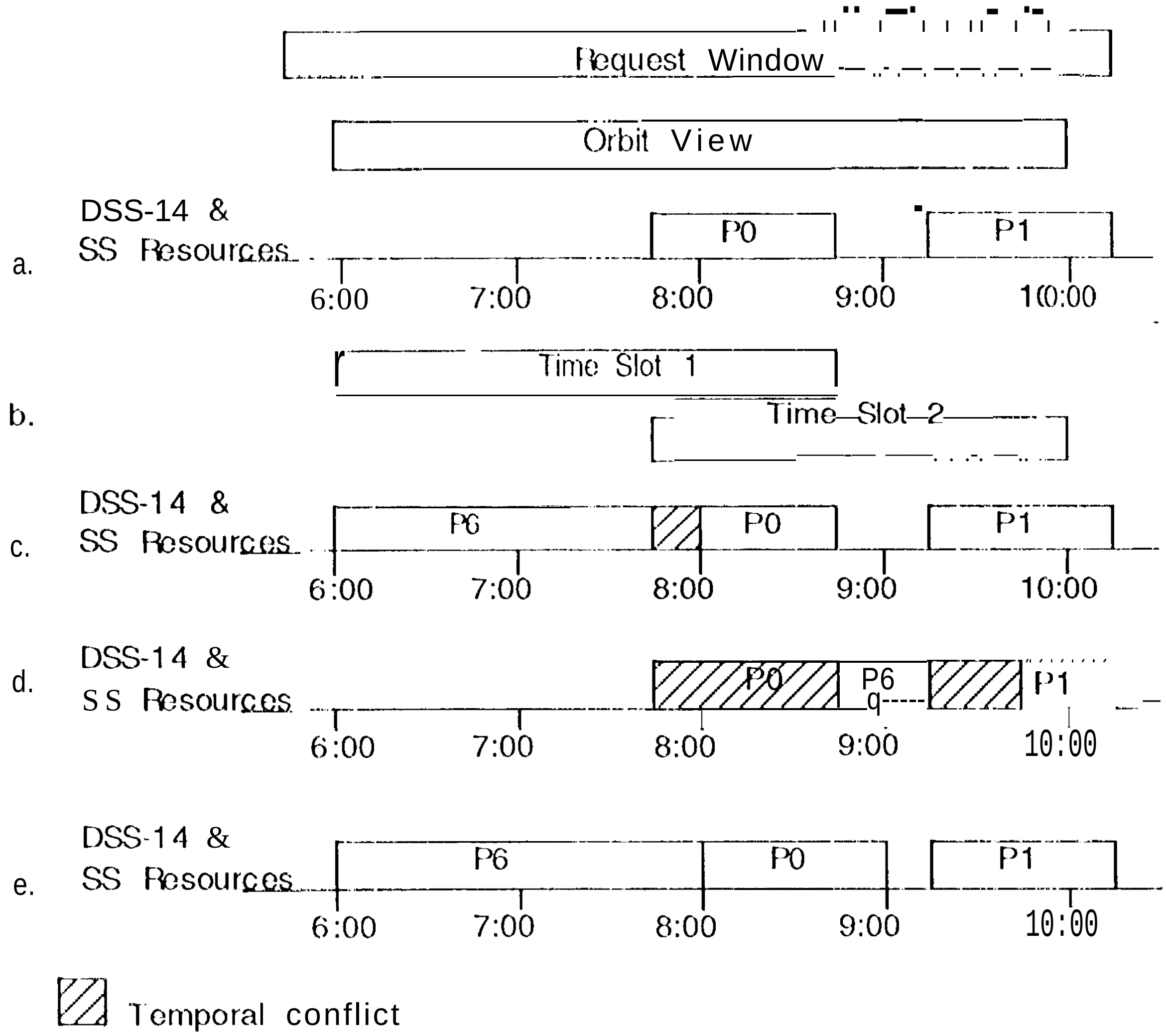


Figure 6: Priority-Based Scheduling Example; a) initial condition; b) valid time slots; c) hypothesis I causes P0 conflicts; d) hypothesis 2 causes P0 & PI conflicts; e) final schedule after placing P6 at 6:00am and rescheduling 1'0 to 8: 00am.

can occur in two ways: (1) it can be initiated top-down due to a change to a previously scheduled track or addition of another request; and (2) it can occur bottom-up in that equipment outages can occur or tracks can fail necessitating rescheduling. In the event of a new or modified request, DANS uses localized search to consider alternative methods for satisfying the new request. This search uses as its bounding function a disruption cost measure which accounts for the overhead involved in moving already scheduled tracks and also a satisfaction measure accounting for what level of requests have been satisfied. Because we use branch and bound techniques DANS can guarantee that it will provide a reschedule optimal with respect to the combined disruption and satisfaction cost function.

In the event of a change in equipment availability, we. are currently examining two solution methods. In both methods DANS first updates all resource timelines to reflect the new resource level. Then, depending on the. size of the change there are (wo options. First, if the change is localized DANS can perform branch and bound search to re-evaluate requests in light of the new equipment situation. However, if the change is to large in scope this search is intractable. For example, if an antenna unexpectedly goes down for a several day period the cascading effect on tracks can be quite great and thus rule out exhaustive search techniques. In these cases DANS can instead first performs prioritized pre-emption to remove low-priority tracks to remove conflicts (by removing the lowest priority tracks participating in each conflict) and then re-evaluate project requests. This approach requires far less search but can produce suboptimal results (with respect to the twin goals of minimizing disruption and maximizing request satisfaction). In the previously discussed taxonomy of reasons for complexity in execution and replanning, this type of rescheduling corresponds to dynamism.

9. CONCLUSIONS

This paper has described the Demand-based Automated Network Scheduling system (DANS), an automated scheduling system being developed at the Jet Propulsion Laboratory (JPL) to schedule DSN resources. DANS uses localized search and priority-based pre-emption to perform priority-based rescheduling in response to changing network demand. In this techniques, DANS first considers the antenna allocation process, as ante.tmas are the central focus of resource contention. After establishing a range of antenna options, DANS then considers allocation of the 5-13 subsystems per track (out of the tens of shard subsystems at each antenna complex) used by each track. DANS uses localized priority-driven, best first search to efficiently consider the large set of possible subsystems schedules.

REFERENCES

[1] M. Deale, M. Yvanovich, D. Schnitzius, D. Kautz, M. Carpenter, M. Zweben, G. Davis, and B. Daun, "The Space Shuttle Ground Processing System," in *Intelligent Scheduling*, Morgan Kaufman, San Francisco, 1994.

[2] Deep Space Network, Jet Propulsion Laboratory Publication 400-517, April 1994.

[3] T. Estlin, X. Wang, A. Govindjee, and S. Chien, "DPLAN Deep Space Network Antenna Operations Planner Programmers Guide Version 1.0," JPL Technical Document D-13377, Jet Propulsion Laboratory, California Institute of Technology, February 1996.

[3] G. Fox and C. Borden, "Low Earth Orbiter Resource Allocation and Cpacity Planning for the Deep Space Network Using LEO4CAST," *Telecommunications and Data Acquisition* 42-118, pp. 169-178, August 1994.

[4] R. W. Hill, Jr., S. A. Chien, K. V. Fayyad, C. Smyth, T. Santos, and R. Bevan, "Sequence of Events Driven Automation of the Deep Space Network," *Telecommunications and Data Acquisition* 42-124, October-December 1995.

[5] M. Johnston & S. Minton, "Analyzing a Heuristic Strategy for Constraint Satisfaction and Scheduling," in *Intelligent Scheduling*, Morgan Kaufman, San Francisco, 1994.

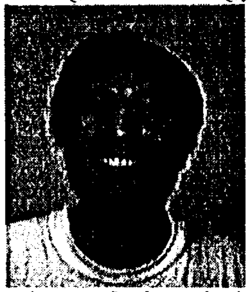
[6] E. Kan, J. Rosas, and Q. Vu, "Operations Mission Planner – 26M Users Guide Modified 1.0", JPL Technical Document D-10092, April 1996

[7] S. J. Loyola, "PC4CAST - A Tool for Deep Space Netowrk Load Forecasting and Capacity Planning," *Telecommunications and Data Acquisition*, 42-114, pp. 170-194, August 1993.

[8] M. Zweben, B. Daun, E. Davis, & M. Deale, Scheduling and Rescheduling with Iterative Repair, in *Intelligent Scheduling*, Morgan Kaufman, San Francisco, 1994.

***Steve Chien** is Technical Group Supervisor of the Artificial Intelligence Group of the Jet Propulsion Laboratory, California Institute of Technology where he leads efforts in research and development of automated planning and scheduling systems. He is also an adjunct assistant professor in the Department of Computer Science at the University of Southern California. He holds a B.S., M.S., and Ph.D. in Computer Science from the University of Illinois. His research interests are in the areas of: planning and scheduling, operations research, and machine learning.*

Ray Lam is a member of' the technical staff of' the Artificial Intelligence Group of the Jet Propulsion Laboratory, California Institute 0s Technology where he participates on the K'search and developent 01 automated scheduling systems. He holds a 11. S. degree in 1 Electrical Engineering from University of California, Los Angeles; and a M.S. degree in Computer Engineering from University of Southern California. His research interests are in the areas of: scheduling, operations research, and neural networks.



Quoc Vu is a member of the technical staff of the Automation and Scheduling Technology Group of the Jet Propulsion Laboratory, California Institute of Technology where he participates on the research and development of automated scheduling systems. He holds a 11, S. degree in ('01111puter Science from California Polytechnic University, San Luis Obispo; and a M.S. degree in Computer Science from Asuza Pacific University, California. His research interests are in the areas of: planning and scheduling, operations research, and graphical user interface.